Towards Scalable and Accurate Online Feature Selection for Big Data

Kui Yu[1], Xindong Wu[2, 3], Wei Ding[4], and Jian Pei[1]

[1]School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
[2]Department of Computer Science, Hefei University of Technology, Hefei, Anhui Province, China
[3]Department of Computer Science, University of Vermont, Burlington, VT, USA
[4]Department of Computer Science, University of Massachusetts Boston, Boston, MA, USA

kuiy@sfu.ca, xwu@uvm.edu, ding@cs.umb.edu, jpei@cs.sfu.ca

***Abstract*—Feature selection is important in many big data applications. There are at least two critical challenges. Firstly, in many applications, the dimensionality is extremely high, in millions, and keeps growing. Secondly, feature selection has to be highly scalable, preferably in an online manner such that each feature can be processed in a sequential scan. In this paper, we develop SAOLA, a Scalable and Accurate OnLine Approach for feature selection. With a theoretical analysis on a low bound on the pairwise correlations between features in the currently selected feature subset, SAOLA employs novel online pairwise comparison techniques to address the two challenges and maintain a parsimonious model over time in an online manner. An empirical study using a series of benchmark real data sets shows that SAOLA is scalable on data sets of extremely high dimensionality, and has superior performance over the state-of-the-art feature selection methods.**

Keywords*—*Online feature selection, Feature redundancy, Extremely high dimensionality

I. Introduction

In the era of big data, many novel applications, such as social media services, high resolution images, genomic data analysis, and document data analysis, consume data of extremely high dimensionality, in the order of millions. For example, the Web Spam Corpus 2011 [13] collected approximately sixteen million features (attributes) for web spam page detection, and the data set from KDD CUP 2010 about using educational data mining to accurately predict student performance includes more than twenty-nine million features. To handle millions of features, the scalability of feature selection methods becomes critical.

Moreover, in many applications, feature selection has to be conducted in an online manner. For example, in SINA Weibo, hot topics in Weibo keep changing daily. When a novel hot topic appears, it may come with a set of new keywords (or a set of features). And then some of the new keywords may serve as key features to identify the new hot topics. Another example is feature selection in bioinformatics, where acquiring the full set of features for every training instance is expensive because of the high cost in conducting wet lab experiments [14]. Accordingly in some real-world applications, it is impossible to wait for a complete set of features. Instead, it is important to conduct feature selection from the features available so far, and consume new features in an online manner as they become available.

Feature selection on high-dimensional data has been generally viewed as a problem of searching for a minimal subset of features that leads to the most accurate prediction model [4], [7], [12], [15]. Two types of feature selection approaches were proposed in the literature, namely batch methods and online methods [2], [10], [14], [16].

A batch method has to access the entire feature set on the training data and performs a global search for the best feature at each round [2]. Accordingly, batch methods cannot be highly scalable for high dimensional data applications that require online feature selection.

Contrast to the batch methods, online (also known as streaming) feature selection is a relative new direction, such as the Fast-OSFS and alpha-investing algorithms [16], [21]. Such a method assumes that features arrive one at a time, and maintains a best feature subset from the features seen so far by processing each feature upon its arrival. Although there is encouraging progress by the existing online feature selection methods, they still meet difficulty in computational cost when the dimensionality is in the scale of millions or more [16].

In this paper, we tackle the challenges in online feature selection from extremely high dimensional data, and develop SAOLA, a Scalable and Accurate OnLine Approach for feature selection. More specifically, to process each new feature efficiently, we have a theoretical analysis to derive a low bound on pairwise correlations between features so that we can filter out redundant features. With this theoretical analysis, the SAOLA algorithm employs novel online pairwise comparisons to address the two challenges and maintain a parsimonious model over time in an online manner. An empirical study using a series of benchmark data sets illustrates that our method is scalable on data sets of extremely high dimensionality, and has superior performance over the state-of-the-art online feature selection methods.

The rest of the paper is organized as follows. Section II presents the preliminaries and reviews related work. Section III proposes our SAOLA algorithm, and Section IV reports our experimental results. Finally, Section V concludes the paper.

II. Preliminaries and Related Work

Given a set F of input features on a training data set, the problem of feature selection is to select a subset of relevant features from F without performance degradation of prediction models. The features in F can be categorized into three disjoint groups, namely, strongly relevant features, weak relevant features, and irrelevant features [5].

Yu and Liu [18] further divided weakly relevant features into redundant and non-redundant features based on Markov blankets.

Definition 1 (Markov Blankets) [6] A Markov blanket of feature F_i, denoted as $M \subseteq F - \{F_i\}$ makes all other features independent of F_i given M, that is,

$$\forall Y \in F - (M \cup \{F_i\}) \; s.t. \; P(F_i|M, Y) = P(F_i|M).\square$$

Definition 2 (Redundant Features) A feature $F_i \in F$ is a redundant feature and hence should be discarded from F, if it has a Markov blanket within F.□

Accordingly, a desirable feature selection method should select strongly relevant features and non-redundant features from input features [5], [18], such as the well-established mRMR (minimal-Redundancy-Maximal-Relevance) algorithm [9] and the FCBF (Fast Correlation-Based Filter) algorithm [18]. Recently, Brown et al. [2] unified almost two decades of research on heuristic scoring criteria for information theoretic feature selection into a new framework using an optimization of the conditional likelihood as a novel interpretation of information theoretic feature selection. Zhao et al. [20] proposed a novel framework to consolidate different criteria to handle feature redundancies. To tackle a huge number of features, Tan et al. [11] proposed the FGM (Feature Generating Machine) algorithm, and Zhai et al. [19] further presented the efficient GDM (Group Discovery Machine) algorithm that outperforms the FGM algorithm.

Since the batch methods have to access all features before feature selection starts, they cannot be easily scalable for high dimensional data analytics that calls for online feature selection.

Contrast to the batch methods, recently, Wang et al. [14] proposed an online feature selection method, OFS, which assumes data instances are sequentially presented, and performs feature selection upon each data instance's arrival. Different from OFS, Zhou et al. [21] presented Alpha-investing which sequentially considers new features as the addition to a predictive model by modeling the candidate feature set as a dynamically generated stream. However, Alpha-investing requires the prior information of the original feature set and never evaluates the redundancy among the selected features as time goes. To tackle the drawbacks, Wu et al. [16] presented the OSFS (Online Streaming Feature Selection) algorithm and its faster version, the Fast-OSFS algorithm. However, facing the scalability and online processing challenges in big data analytics, the computational cost inherent in those three algorithms may still be prohibitive when the dimensionality is extremely high in the scale of millions or more.

Accordingly, those challenges motivate us to develop a scalable and online processing method to deal with data with extremely high dimensionality.

III. The proposed Algorithm

A. Problem Definition

Given a training data set $D = \{(d_i, c_i), 1 \leq i \leq N\}$, where N is the number of data instances, each data instance d_i is a multidimensional vector that contains P features, and C is the class attribute that has K distinct class labels, $c_i \in \{c_1, c_2, ..., c_K\}$.

We also denote D by $D = \{(F_i, C), 1 \leq i \leq P\}$, which is a sequence of features that is presented in a sequential order, where $F_i = \{f_1, f_2, ..., f_N\}^T$ denotes the i^{th} feature containing N data instances, and C includes N class label instances.

If D can be processed in a sequential scan, that is, one dimension at a time, we can process high dimensional data not only with limited memory, but also without requiring its complete set of features available. The challenge is that, as we process one dimension at a time, at any time t_i, how to online maintain a minimum feature subset $S^\star_{t_i}$ of maximizing its predictive performance for classification. Assuming $S \subseteq F$ is the feature set containing all features available till time t_{i-1} and F_i is a new coming feature at time t_i, our problem can be formulated as follows:

$$S^\star_{t_i} = \arg\min_{S'} \{|S'| : S' = \arg\max_{\zeta \subseteq \{S \cup F_i\}} P(C|\zeta)\}. \quad (1)$$

We can further decompose it into the following key steps:

- Determine the relevance of F_i to C. Firstly, we determine whether Eq.(2) holds or not.

$$P(C|F_i) = P(C). \quad (2)$$

  If so, F_i is discarded as an irrelevant feature. If not, secondly, we further evaluate whether F_i carries additional predictive information to C given the selected feature set $S^\star_{t_{i-1}}$ at t_{i-1}, that is, whether Eq.(3) holds. If Eq.(3) holds, F_i will be discarded.

$$P(C|S^\star_{t_{i-1}}, F_i) = P(C|S^\star_{t_{i-1}}). \quad (3)$$

- Calculate $S^\star_{t_i}$ with $F_i's$ inclusion. Once F_i is added to $S^\star_{t_{i-1}}$, at time t_i, S_{t_i}=$\{S^\star_{t_{i-1}}, F_i\}$, we then solve Eq.(4) to prune S_{t_i} to satisfy Eq.(1).

$$S^\star_{t_i} = \arg\max_{\zeta \subseteq S_{t_i}} P(C|\zeta)\}. \quad (4)$$

Accordingly, solving Eq.(1) is decomposed to how to sequentially solve Eq.(2) to Eq.(4) at each time point. Essentially, Eq.(3) and Eq.(4) deal with the problem of feature redundancy. We can apply Definition 2 in Section 2 to solve Eq.(3) and Eq.(4). However, it is computationally expensive to use Definition 2 when the number of features within $S^\star_{t_{i-1}}$ is large. To evaluate whether F_i is redundant with respect to $S^\star_{t_{i-1}}$ using the standard Markov blanket filtering criterion (Definitions 1 and 2), it is necessary to check all the subsets of $S^\star_{t_{i-1}}$ (the total number of subsets is $2^{|S^\star_{t_{i-1}}|}$) to determine which subset subsumes the predictive information that F_i has about C. If such a subset is found, F_i becomes redundant and is removed. When handling a larger number of features, it is computationally prohibitive to check all the subsets of $S^\star_{t_{i-1}}$.

Methods such as greedy search are a natural fit for this problem setting. In [16], a k-greedy search strategy is adopted to evaluate redundant features. It checks all subsets of size less than or equal to ι ($1 \leq \iota \leq |S^{\star}_{t_{i-1}}|$), where ι is a user-defined parameter. However, when the size of $S^{\star}_{t_{i-1}}$ is large, it is still computationally prohibitive to evaluate the subsets of size up to ι. Moreover, selecting a proper value of ι is difficult. Therefore, those challenges motivate us to develop a scalable and online processing method to solve Eq.(3) and Eq.(4) for big data analytics.

B. The Solutions to Eq.(2), Eq.(3), and Eq.(4)

In this section, to cope with computational complexity, we propose pairwise comparisons to online calculate the correlations between features, instead of computing the correlation between features conditioned on all the feature subsets. For pairwise comparisons, we employ the measure of mutual information to calculate correlations between features. Given two variables Y and Z, the mutual information between Y and Z is defined as follows.

$$I(Y;Z) = H(Y) - H(Y|Z). \tag{5}$$

The entropy of a feature Y is defined as

$$H(Y) = -\Sigma_{y_i \in Y} P(y_i) \log_2 P(y_i). \tag{6}$$

And the entropy of Y after observing values of another variable Z is defined as

$$H(Y|Z) = -\Sigma_{z_j \in Z} P(z_j) \Sigma_{y_i \in Y} P(y_i|z_i) \log_2 P(y_i|z_i), \tag{7}$$

where $P(y_i)$ is the prior probability of value y_i of variable Y, and $P(y_i|z_i)$ is the posterior probability of y_i given the value z_i of variable Z.

With mutual information as a correlation measure between features, we propose solutions to Eq.(2), Eq.(3), and Eq.(4) as follows.

1) The Solution to Eq.(2): Assuming $S^{\star}_{t_{i-1}}$ is the selected feature subset at time t_{i-1}, and at time t_i, a new feature F_i comes, to solve Eq.(2), given a relevance threshold δ_1, if $I(F_i, C) > \delta_1$ ($0 \leq \delta_1 \leq 1$), F_i is said to be a relevant feature to C; otherwise, F_i is discarded as an irrelevant feature and will never be considered again.

2) The Solution to Eq.(3): If F_i is a relevant feature, at time t_i, how can we determine whether F_i should be kept given $S^{\star}_{t_{i-1}}$, that is, whether $I(C; F_i|S^{\star}_{t_{i-1}}) = 0$? If $\exists Y \in S^{\star}_{t_{i-1}}$ such that $I(F_i; C|Y) = 0$, it testifies that adding F_i alone to $S^{\star}_{t_{i-1}}$ does not increase the predictive capability of $S^{\star}_{t_{i-1}}$. With this observation, we solve Eq.(3) with the following lemmas.

Lemma 1 With the current feature subset $S^{\star}_{t_{i-1}}$ at time t_{i-1} and a new feature F_i at time t_i, if $\exists Y \in S^{\star}_{t_{i-1}}$ such that $I(F_i; C|Y) = 0$, then $I(F_i; Y) \geq I(F_i; C)$.

Proof. Making use of the identity, $I(F_i; C|Y) - I(F_i; C) = I(F_i; Y|C) - I(F_i; Y)$, we get Eq.(8) as follows.

$$I(F_i; C|Y) = I(F_i; C) + I(F_i; Y|C) - I(F_i; Y). \tag{8}$$

With Eq.(8), if $I(F_i; C|Y) = 0$ holds, we get the following,

$$I(F_i; Y) = I(F_i; C) + I(F_i; Y|C). \tag{9}$$

Using Eq.(9), we get the following bound of $I(F_i; Y)$.

$$I(F_i; Y) \geq I(F_i; C). \square \tag{10}$$

Lemma 1 proposes a correlation bound between features to testify whether a new feature can increase the predictive capability of the current feature subset. Meanwhile, if $I(F_i; C|Y) = 0$ holds, Lemma 2 answers what the relationship between $I(Y; C)$ and $I(F_i; C)$ is.

Lemma 2 With the current feature subset $S^{\star}_{t_{i-1}}$ at time t_{i-1} and a new feature F_i at time t_i, $\exists Y \in S^{\star}_{t_{i-1}}$, if $I(F_i; C|Y) = 0$ holds, then $I(Y; C) > I(F_i; C)$.

Proof. Considering the following identity,

$$I(Y; F_i|C) = I(Y; F_i) + H(Y|C) + H(F_i|C) - H(C|Y, F_i) - H(C),$$

we get $I(Y; F_i|C) = I(F_i; Y|C)$.

With the following relationship and Eq.(9),

$$I(Y; C|F_i) = I(Y; C) + I(Y; F_i|C) - I(F_i; Y),$$

we get the following,

$$I(Y; C|F_i) = I(Y; C) - I(F_i; C).$$

Since Y is in the current feature set $S^{\star}_{t_{i-1}}$, $I(Y; C|F_i) > 0$. Accordingly, the following holds.

$$I(Y; C) > I(F_i; C). \square \tag{11}$$

Theorem 1 With the current feature subset $S^{\star}_{t_{i-1}}$ at time t_{i-1} and a new feature F_i at time t_i, $\exists Y \in S^{\star}_{t_{i-1}}$, if $I(F_i; C|Y) = 0$ holds, then the following is achieved.

$$I(Y; C) > I(F_i; C) \; and \; I(F_i; Y) \geq I(F_i; C). \tag{12}$$

Proof. With Lemmas 1 and 2, Theorem 1 is proved.$\square$

With Theorem 1, we deal with Eq.(3) as follows. With a new feature F_i at time t_i, $\exists Y \in S^{\star}_{t_{i-1}}$, if Eq.(12) holds, then F_i is discarded; otherwise, F_i is added to $S^{\star}_{t_{i-1}}$.

3) The Solution to Eq.(4): Once F_i is added to $S^{*}_{t_{i-1}}$ at time t_i, we will checking which features within $S^{*}_{t_{i-1}}$ can be removed due to the new inclusion of F_i. If $\exists Y \in S^{\star}_{t_{i-1}}$ such that $I(C; Y|F_i) = 0$, then Y is removed from $S^{\star}_{t_{i-1}}$.

Similar to Eq.(8) and Eq.(9), if $I(C; Y|F_i) = 0$, we have $I(Y; F_i) \geq I(Y; C)$. At the same time, if $I(C; Y|F_i) = 0$, similar to Eq.(11), we can get,

$$I(F_i; C) > I(Y; C). \tag{13}$$

With the above analysis and Eq.(13), we get the following,

$$I(F_i; C) > I(Y; C) \; and \; I(Y; F_i) \geq I(Y; C). \tag{14}$$

Accordingly, the solution to Eq.(4) is as follows. With the feature subset $S^{\star}_{t_i}$ at time t_i, and $F_i \in S^{\star}_{t_i}$, if $\exists Y \in S^{\star}_{t_i}$ such that Eq.(14) holds, then Y is removed.

Algorithm 1: *The SAOLA Algorithm*

Data:
F_i : predictive features; C : the class attribute;
δ_1: a relevance threshold ($0 \leq \delta_1 \leq 1$);
δ_2: a correlation bound of $I(F_i;Y)$;
$S^{\star}_{t_{i-1}}$: the selected feature set at time t_{i-1};
$S^{*}_{t_i}$: the selected feature set at time t_i

1 **repeat**
2 Get a new feature F_i at time t_i;
3 /*Solve Eq.(2)*/
4 **if** $I(F_i, C) < \delta_1$ **then**
5 Discard F_i, and go to Step 18;
6 **end**
7 **for** *each feature* $Y \in S^{*}_{t_{i-1}}$ **do**
8 /*Solve Eq.(3)*/
9 **if** $I(Y;C) > I(F_i;C)$ & $I(F_i;Y) \geq \delta_2$ **then**
10 Discard F_i, go to Step 18;
11 **end**
12 /*Solve Eq.(4)*/
13 **if** $I(F_i;C) > I(Y;C)$ & $I(F_i;Y) \geq \delta_2$ **then**
14 $S_{t^{*}_{i-1}} = S_{t^{*}_{i-1}} - Y$;
15 **end**
16 **end**
17 $S^{*}_{t_i} = S_{t^{*}_{i-1}} \cup F_i$;
18 **until** *no features are available*;
19 Output $S^{*}_{t_i}$;

C. The SAOLA Algorithm

Using Theorem 1 and Eq.(14), we propose the SAOLA algorithm in detail, as shown in Algorithm 1.

In Algorithm 1, δ_2 is the correlation bound of $I(F_i;Y)$. According to Eq.(12) and Eq.(14), $\delta_2 = min(I(F_i;C), I(Y;C))$. The SAOLA algorithm is implemented as follows. At time t_i, as a new feature F_i arrives, if $I(F_i, C) < \delta_1$ holds at Step 4, then F_i is discarded as an irrelevant feature and SAOLA waits for a next coming feature; if not, at Step 9, SAOLA evaluates whether F_i should be kept given the current feature set $S^{*}_{t_{i-1}}$. If $\exists Y \in S^{\star}_{t_{i-1}}$ such that Eq.(12) holds, we discard F_i and never consider it again.

Once F_i is added to $S^{*}_{t_{i-1}}$ at time t_i, $S^{*}_{t_{i-1}}$ will be checked whether some features within $S^{*}_{t_{i-1}}$ can be removed due to the new inclusion of F_i. At Step 13, if $\exists Y \in S^{\star}_{t_{i-1}}$ such that Eq.(14) holds, Y is removed.

To reduce computational cost, the SAOLA algorithm proposes a set of pairwise comparisons between individual features instead of conditioning on a set of features, as the selection criterion for choosing features. This is essentially the idea behind the well-established batch feature selection algorithms, such as mRMR and FCBF [9], [18]. Although FCBF proposed a concept of approximate Markov blankets to calculate the correlation between features with pairwise comparisons, it does not give a theoretical analysis on why an approximate Markov blanket works well for feature selection.

The major computation in SAOLA is the computation of the correlations between features (Steps 4 and 9 in Algorithm 1). At time t_i, assuming the total number of features is up to P and $|S^{*}_{t_i}|$ is the number of the currently selected feature set, the time complexity of the algorithm is $O(P|S^{*}_{t_i}|)$. Accordingly, the time complexity of SAOLA is determined by the number of features within $|S^{*}_{t_i}|$. But the strategy of online pairwise comparisons guarantees the scalability of SAOLA, even when the size of $|S^{*}_{t_i}|$ is large.

Comparing to Fast-OSFS, SAOLA employs a k-greedy search strategy to filter out redundant features by checking feature subsets for each feature in $S^{*}_{t_i}$. At time t_i, the best time complexity of Fast-OSFS is $O(|S^{*}_{t_i}|\iota^{|S^{*}_{t_i}|})$, where $\iota^{|S^{*}_{t_i}|}$ denotes all subsets of size less than or equal to ι ($1 \leq \iota \leq |S^{\star}_{t_{i-1}}|$) for checking. With respect to Alpha-investing, at time t_i, the time complexity of Alpha-investing is $O(P|S^{*}_{t_i}|^2)$. Since Alpha-investing only considers adding new features but never evaluates the redundancy of selected features, the feature set $S^{*}_{t_i}$ always has a large size. Thus, when the size of candidate features is extremely high and the size of $|S^{*}_{t_i}|$ becomes large, Alpha-investing and Fast-OSFS both become computationally intensive or even prohibitive. Moreover, how to select a suitable value of ι for Fast-OSFS in advance is a hard problem, since different data sets may require different ι to search for a best feature subset.

Finally, for data with discrete values, we use the measure of mutual information, while for data with discrete values, we adopt the best known measure of the Fisher's Z-test [8] to calculate correlations between features. In a Gaussian distribution, $Normal(\mu, \Sigma)$, the population partial correlation $p_{(XY|S)}$ between feature X and feature Y given a feature subset S is calculated as follows.

$$p_{(XY|S)} = \frac{-((\sum_{XYS})^{-1})_{XY}}{((\sum_{XYS})^{-1})_{XX}((\sum_{XYS})^{-1})_{YY}} \qquad (15)$$

In the Fisher's Z-test, under the null hypothesis of the conditional independence between X and Y given S that $p_{(XY|S)} = 0$. With the Fisher's Z-test, assuming α is a given significance level and ρ is the p-value returned by the Fisher's Z-test, under the null hypothesis of the conditional independence between X and Y, X and Y are uncorrelated to each other, if $\rho > \alpha$; otherwise, X and Y are correlated to each other, if $\rho \leq \alpha$. Accordingly, at time t, a new feature F_i correlated to C is discarded given $S^{*}_{t_{i-1}}$, if $\exists Y \in S^{*}_{t_{i-1}}$ s.t. $p_{Y,C} > p_{F_i,C}$ and $p_{Y,F_i} > p_{F_i,C}$.

IV. EXPERIMENT RESULTS

A. Experiment Setup

We use fourteen benchmark data sets as our test beds, including ten high-dimensional data sets [1], [17] and four extremely high-dimensional data sets, as shown in Table I. The first ten high-dimensional data sets include two biomedical data sets (*hiva* and *breast-cancer*), three NIPS 2003 feature selection challenge data sets (*dexter*, *madelon*, and *dorothea*), and two public microarray data sets (*lung-cancer* and *leukemia*), two massive high-dimensional text categorization data sets (*ohsumed* and *apcj-etiology*), and the *thrombin* data set that is chosen from KDD Cup 2001. The last four data sets with extremely high dimensionality are available at the Libsvm data set website[1].

[1] http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/

In the first ten high-dimensional data sets, we use the originally provided training and validation sets for the three NIPS 2003 challenge data sets and the *hiva* data set, and for the remaining six data sets, we adopt 2/3 instances for training and the remaining 1/3 instances for testing. In the *news20* data set, we use the first 9996 data instances for training and the rest for testing while in the *url1* data set, we use the first day data set (*url1*) for training and the second day data set (*url2*) for testing. In the *kdd2010* and *webspam* data sets, we randomly select 20000 data instances for training, and 100,000 and 78,000 data instances for testing, respectively. Our comparative study compares the SAOLA algorithm with the following algorithms:

- Three state-of-the-art online feature selection methods: Fast-OSFS [16], Alpha-investing [21], and OFS [14]. Fast-OSFS and Alpha-investing assume features on training data arrive one by one at a time while OFS assumes data examples come one by one;
- Three batch methods: one well-established algorithm of FCBF [18], and two state-of-the-art algorithms, SPSF-LAR [20] and GDM [19].

TABLE I. THE BENCHMARK DATA SETS

Dataset	♯ features	♯training instances	♯testing instances
madelon	500	2,000	600
hiva	1,617	3,845	384
leukemia	7,129	48	24
lung-cancer	12,533	121	60
ohsumed	14,373	3,400	1,600
breast-cancer	17,816	190	96
dexter	20,000	300	300
apcj-etiology	28,228	11,000	4,779
dorothea	100,000	800	300
thrombin	139,351	2,000	543
news20	1,355,191	9,996	10,000
url1	3,231,961	20,000	20,000
webspam	16,609,143	20,000	78,000
kdd2010	29,890,095	20,000	100,000

We use two classifiers, KNN and J48 provided in the Spider Toolbox[2] to evaluate a selected feature subset in the experiments. All experiments were conducted on a computer with Interl(R) i7-2600, 3.4GHz CPU, and 24GB memory. In the remaining sections, the parameter δ_1 for SAOLA is set to 0 for discrete data while the significance level α for SAOLA is set to 0.01 for the Fisher's Z-test for continuous data.

B. Comparison of SAOLA with Three Online Algorithms

1) Comparison of SAOLA with Fast-OSFS and Alpha-investing: Since Fast-OSFS and Alpha-investing can only deal with the first ten high-dimensional data sets in Table I due to high computational cost, in this section we compare them with SAOLA in terms of prediction accuracy, size of selected feature subsets, and running time on the first ten high-dimensional data sets. The significance level is set to 0.01 for Fast-OSFS, and for Alpha-investing, the parameters are set to the values used in [21].

Tables II and III summarize the prediction accuracies of SAOLA against Fast-OSFS and Alpha-investing using the KNN and J48 classifiers. We conduct paired t-tests at a 95% significance level and summarize the win/tie/lose (w/t/l for short) counts of SAOLA against Fast-OSFS and Alpha-investing in the last rows of Tables II and III. The highest prediction accuracy is highlighted in bold face. Table IV gives the number of selected features of SAOLA, Fast-OSFS, and Alpha-investing. We have the following observations.

[2]http://people.kyb.tuebingen.mpg.de/spider/

TABLE II. PREDICTION ACCURACY (J48)

Dataset	SAOLA	Fast-OSFS	Alpha-investing
dexter	0.8133	**0.8200**	0.5000
lung-cancer	**0.9500**	0.9000	0.8333
hiva	**0.9661**	0.9635	0.9635
breast-cancer	0.6042	0.6771	**0.7188**
leukemia	**0.9583**	**0.9583**	0.6667
madelon	0.6083	**0.6100**	0.6067
ohsumed	0.9437	**0.9450**	0.9331
apcj-etiology	**0.9872**	0.9868	0.9828
dorothea	0.9343	**0.9371**	0.9343
thrombin	**0.9613**	0.9595	**0.9613**
average rank	2.25	**2.30**	1.45
w/t/l	-	1/8/1	4/5/1

TABLE III. PREDICTION ACCURACY (KNN)

Dataset	SAOLA	Fast-OSFS	Alpha-investing
dexter	0.7600	**0.7800**	0.5000
lung-cancer	**0.9833**	0.9667	0.9167
hiva	**0.9635**	**0.9635**	0.9531
breast-cancer	**0.6771**	0.6667	0.5833
leukemia	**0.9167**	0.7917	0.6250
madelon	0.5617	0.5283	**0.5767**
ohsumed	0.9275	0.9306	**0.9325**
apcj-etiology	0.9793	0.9702	**0.9851**
dorothea	**0.9613**	0.9457	0.7400
thrombin	**0.9374**	0.9300	0.9371
average rank	**2.45**	1.85	1.70
w/t/l	-	5/4/1	6/3/1

(1) SAOLA vs. Fast-OSFS. With the counts of win/tie/loss (w/t/l) in Table II, we observe that SAOLA is very competitive with Fast-OSFS. In Table III, we can see that SAOLA is superior to Fast-OSFS. Fast-OSFS selects fewer features than SAOLA on all data sets as shown in Table IV. The explanation is that Fast-OSFS employs a k-greedy search strategy to filter out redundant features by checking all feature subsets for each feature in the current feature set while SAOLA only uses pairwise comparisons. But this strategy makes Fast-OSFS very expensive in computation and even prohibitive on some data sets, such as *apcj-etiology* and *thrombin* as shown in Table V, as the size of the current feature set is large at each time point.

(2) SAOLA vs. Alpha-investing. With Tables II and III, we can see that SAOLA outperforms Alpha-investing on most data sets using the KNN and J48 classifiers. Alpha-investing selects many more features than SAOLA on the last four data sets in Table IV, since Alpha-investing only considers to add new features but never evaluates the redundancy of selected features. An exception is that Alpha-investing only selects one feature on the *dexter* data set. A possible explanation is that the *dexter* data set is a very sparse real-valued data set. Furthermore, Alpha-investing is less efficient than SAOLA as shown in Table V.

To validate whether SAOLA, Fast-OSFS, and Alpha-investing have no significant difference in prediction accuracy, with the Friedman test at 95% significance level [3], under the null-hypothesis, which states that the performance of SAOLA and that of Fast-OSFS and Alpha-investing have no significant difference, for the KNN classifier, the average ranks calculated from the Friedman test for SAOLA, Fast-OSFS, and Alpha-

TABLE IV. NUMBER OF SELECTED FEATURES

Dataset	SAOLA	Fast-OSFS	Alpha-investing
dexter	21	9	1
lung-cancer	35	6	7
hiva	12	5	48
breast-cancer	46	7	2
leukemia	17	5	2
madelon	3	3	4
ohsumed	65	11	297
apcj-etiology	75	67	634
dorothea	63	5	113
thrombin	20	9	60

investing are 2.45, 1.85, and 1.70 (the higher the average rank, the better the performance) in Table III, respectively. Meanwhile, with respect to J48, the average ranks for SAOLA, Fast-OSFS, and Alpha-investing are 2.25, 2.30, and 1.45 in Table II (how to calculate the average ranks, please see [3]), respectively.

In summary, in prediction accuracy, SAOLA is very competitive, or even better than Fast-OSFS, and is superior to Alpha-investing. Furthermore, Fast-OSFS and Alpha-investing cannot deal with extremely high-dimensional data sets due to computational cost while SAOLA is accurate and scalable.

TABLE V. RUNNING TIME (SECONDS)

Dataset	SAOLA	Fast-OSFS	Alpha-investing
dexter	**3**	4	6
lung-cancer	6	4	**2**
hiva	**1**	36	7
breast-cancer	5	4	**3**
leukemia	2	2	**1**
madelon	**0.1**	**0.1**	**0.1**
ohsumed	**6**	343	497
apcj-etiology	**22**	$>$ 3 days	9,781
dorothea	**58**	375	457
thrombin	**63**	18,576	291

2) Comparison of SAOLA with OFS: The OFS algorithm is a recently proposed online feature selection method. Since OFS uses a user-defined parameter k to control the size of the final selected feature subset, we set the parameter for the number of selected features as follows: (1) OFS1, selecting the same number of features as the SAOLA algorithm; (2) OFS2, setting the user-defined parameter k, i.e., the number of selected features to the top 5, 10, 15 ,..., 100 features, then selecting the feature set with the highest prediction accuracy as the reporting result.

TABLE VI. PREDICTION ACCURACY (KNN)

Dataset	SAOLA	OFS1	OFS2
dexter	**0.7600**	0.4700	0.5400
lung-cancer	**0.9833**	0.7500	0.8500
hiva	0.9635	**0.9661**	**0.9661**
breast-cancer	**0.6771**	0.5938	0.6667
leukemia	**0.9167**	0.7500	0.8750
madelon	0.5617	0.5183	**0.6433**
ohsumed	0.9275	0.9287	**0.9431**
apcj-etiology	0.9793	0.9835	**0.9872**
dorothea	**0.9613**	0.8000	0.9086
thrombin	0.9374	0.9263	**0.9411**
news20	0.7755	**0.8423**	0.6884
url1	0.9627	**0.9757**	0.9607
kdd10	**0.8780**	0.8527	0.7755
webspam	0.9532	**0.9650**	0.9516
average rank	**2.2143**	1.7500	2.0357
w/t/l	-	8/3/3	7/5/2

With Tables VI and VII, to evaluate whether the performance of SAOLA and that of OSF1 and OFS2 have no significant difference in prediction, we use the Friedman test at 95% significance level under the null-hypothesis, which states that the performance of SAOLA and that of OSF1 and OFS2 have no significant difference in prediction accuracy. For the J48 classifier, the null-hypothesis is rejected, and the average ranks for SAOLA, OSF1, and OFS2 are 2.4643, 1.3929, 2.1429, respectively.

TABLE VII. PREDICTION ACCURACY (J48)

Dataset	SAOLA	OFS1	OFS2
dexter	**0.8133**	0.5600	0.5667
lung-cancer	**0.9500**	0.7667	0.8667
hiva	**0.9661**	0.9635	0.9635
breast-cancer	0.6042	0.6458	**0.6563**
leukemia	**0.9583**	0.7500	**0.9583**
madelon	0.6083	0.5600	**0.6367**
ohsumed	**0.9437**	0.9431	0.9431
apcj-etiology	**0.9872**	**0.9872**	**0.9872**
dorothea	0.9343	0.9314	**0.9371**
thrombin	**0.9613**	0.9263	0.9374
news20	**0.8276**	0.7757	0.7332
url1	**0.9744**	0.9027	0.9720
kdd10	**0.8723**	0.8532	0.8577
webspam	0.9611	**0.9689**	**0.9689**
average rank	**2.4643**	1.3929	2.1429
w/t/l	-	8/5/1	5/7/2

Then we proceed with the Nemenyi test [3] as a post-hoc test. With the Nemenyi test, the performance of two methods is significantly different if the corresponding average ranks differ by at least the critical difference (how to calculate the critical difference, please see [3]). With the Nemenyi test, the critical difference is up to 0.8275. Thus, with the critical difference and the average ranks calculated above, the performance of SAOLA and that of OSF2 have no significant difference, but SAOLA is significantly better than OSF1. For KNN, the null-hypothesis cannot be rejected, and the average ranks for SAOLA, OSF1, and OFS2 are 2.2143, 1.7500, 2.0357, respectively. Accordingly, for the KNN classifier, SAOLA, OSF1, and OFS2 have no significant difference in prediction accuracy.

Table VIII gives the running time of SAOLA, OSF1, and OSF2. For OFS2, we record the running time of the feature subset with the highest accuracy as its running time. In Table III, we only give the running time of eight data sets, since on the the remaining six data sets, the running time of SAOLA, OSF1, and OFS2 is no more than five seconds. SAOLA is faster than both OSF1 and OSF2, except for the *dorothea* and *thrombin* data sets. The *dorothea* and *thrombin* data sets only include 800 samples and 2000 samples, respectively. When the number of data samples becomes large and the number of features of training data is increased to millions, OFS1 and OFS2 become very costly, and SAOLA is still scalable and efficient. The explanation is that the time complexity of SAOLA is determined by the number of features within the currently selected feature set, and the strategy of online pairwise comparisons makes SAOLA very scalable, even when the size of the current feature set is large. Moreover, setting a desirable size of a feature set selected by OFS2 in advance is a non-trivial task.

Figure 1 shows the number of selected features in SAOLA and OSF2. OSF1 is set to select the same features as SAOLA. Thus we do not plot OSF1 in Figure 1. We can see that SAOLA selects fewer features than OSF2 on all data sets except for *breast-cancer*, *ohsumed*, *acpj*, *news20*, and *kdd10*.

TABLE VIII. RUNNING TIME (SECONDS)

Dataset	SAOLA	OFS1	OFS2
ohsumed	**6**	9	9
apcj-etiology	**22**	77	100
dorothea	58	**7**	10
thrombin	63	**36**	40
news20	**944**	1910	1572
url1	**200**	1234	1837
kdd10	**1056**	26793	28536
webspam	**1456**	20127	18342

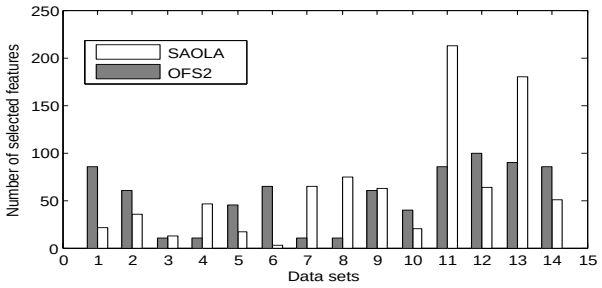

Fig. 1. Number of selected features (The labels of the x-axis from 1 to 10 denote the data sets: 1. dexter; 2. lung-cancer; 3. hiva; 4. breast-cancer; 5. leukemia; 6. madelon; 7. ohsumed; 8. apcj-etiology; 9. dorothea; 10. thrombin; 11. news20; 12. url1; 13. kdd10; 14. webspam)

TABLE IX. PREDICTION ACCURACY (J48)

Dataset	SAOLA	FCBF	SPSF-LAR
dexter	0.8133	0.8567	**0.8700**
lung-cancer	0.9500	0.9500	**0.9833**
hiva	**0.9661**	**0.9661**	0.9635
breast-cancer	0.6042	0.6042	**0.6458**
leukemia	**0.9583**	**0.9583**	**0.9583**
madelon	0.6083	0.6067	**0.6183**
ohsumed	0.9437	**0.9444**	0.9431
apcj-etiology	**0.9872**	0.9866	**0.9872**
dorothea	**0.9343**	0.9314	0.9029
thrombin	**0.9613**	0.9576	0.9558
average rank	1.85	**2.15**	2.00
w/t/l	-	0/9/1	1/5/4

TABLE X. PREDICTION ACCURACY (KNN)

Dataset	SAOLA	FCBF	SPSF-LAR
dexter	0.7600	**0.7967**	0.7233
lung-cancer	**0.9833**	0.9500	**0.9833**
hiva	**0.9635**	0.9609	**0.9635**
breast-cancer	**0.6771**	0.6563	**0.6771**
leukemia	0.9167	**1.0000**	**1.0000**
madelon	0.5617	**0.5767**	0.5633
ohsumed	0.9275	**0.9300**	0.9113
apcj-etiology	0.9793	**0.9826**	0.9803
dorothea	**0.9613**	0.9200	0.8857
thrombin	0.9374	0.9429	**0.9650**
average rank	**2.10**	1.85	2.05
w/t/l	-	3/4/3	3/5/2

C. Comparison with Three Batch Methods

1) Comparison with FCBF and SPSF-LAR: Since FCBF and SPSF-LAR can only deal with the first ten high-dimensional data sets in Table I, in this section we compare them with our proposed algorithm in terms of prediction accuracy, size of selected feature subsets, and running time. The information threshold for FCBF is set to 0. We set the user-defined parameter k, i.e., the number of selected features to the top 5, 10, 15 ,..., 65 features for the SPSF-LAR algorithm, and choose the feature subsets of the highest prediction accuracy.

Tables IX and X report the prediction accuracies of SAOLA against FCBF and SPSF-LAR. With the counts of win/tie/loss (w/t/l) in the last rows of Tables IX and X, we can see that even without requiring the entire feature set on a training data set in advance, SAOLA is still very competitive with both FCBF and SPSF-LAR in prediction accuracy.

To further validate whether the performance of SAOLA is comparable to that of FCBF and SPSF-LAR in prediction accuracy, we use the Friedman test at 95% significance level. For the KNN classifier, the average ranks calculated from the Friedman test for SAOLA, Fast-OSFS, and SPSF-LAR are 2.10, 1.85, and 2.05, respectively. For J48, the average ranks for SAOLA, FCBF and SPSF-LAR are 1.85, 2.15, and 2.00, respectively. Thus, with the Friedman test at 95% significance level, on both KNN and J48, SAOLA, FCBF and SPSF-LAR have no significant difference in prediction accuracy. Accordingly, we conclude that the performance of SAOLA is comparable to that of FCBF and SPSF-LAR.

As for the number of selected features, SPSF-LAR selects the feature set with the highest prediction accuracy from 5, 10, 15 ,..., 65 features, and then we record the running time of this feature set as the running time of SPSF-LAR. In Figure 2, FCBF selects the most features among SAOLA, FCBF and SPSF-LAR while SAOLA and SPSF-LAR are similar to each other. For the running time as shown in Figure 3, SAOLA is the fastest algorithm among SAOLA, FCBF and SPSF-LAR while SPSF-LAR is the slowest.

2) Comparison with the GDM Algorithm: In this section, we select the GDM algorithm [19] which is one of the most recent batch feature selection methods in dealing with very large dimensionality. GDM uses a user-defined parameter to control the size of the final selected feature subset. We set the selected feature subset sizes to the top 10, 20, 30, ..., 260 features for the GDM algorithm, report the running time of the feature subset with the highest accuracy as the running time of GDM, and choose the highest prediction accuracies achieved among those selected feature subsets. Table XI reports the prediction accuracies of SAOLA and GDM. We can see that our algorithm is very competitive with GDM on both J48 and KNN. With the Friedman test on prediction accuracy at 95% significance level, for both KNN and J48, we observe the same average ranks for SAOLA and GDM, 1.3929 and 1.6071, respectively. With the Friedman test at 95% significance level, both Knn and J48 do not have significant difference in prediction accuracy using the feature sets selected by SAOLA and GDM.

Figure 4 shows the running time of SAOLA against GDM.

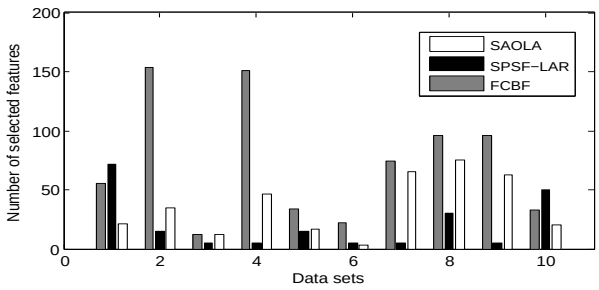

Fig. 2. Number of selected features (The labels of the x-axis from 1 to 10 denote the data sets: 1. dexter; 2. lung-cancer; 3. hiva; 4. breast-cancer; 5. leukemia; 6. madelon; 7. ohsumed; 8. apcj-etiology; 9. dorothea; 10. thrombin)

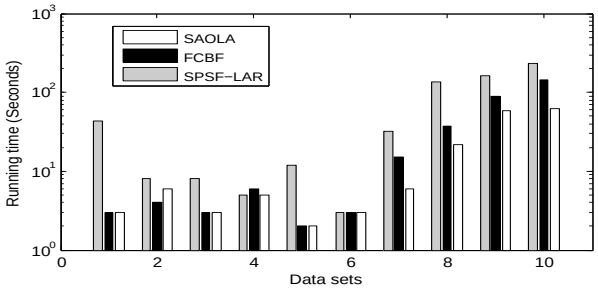

Fig. 3. Running time (The labels of the x-axis are the same as the labels of the x-axis in Figure 2.)

TABLE XI. PREDICTION ACCURACY

Dataset	KNN		J48	
	SAOLA	GDM	SAOLA	GDM
dexter	0.7600	**0.9100**	0.8133	**0.9100**
lung-cancer	**0.9833**	**0.9833**	0.9500	**0.9833**
hiva	0.9635	**0.9661**	**0.9661**	**0.9661**
breast-cancer	**0.6771**	0.4792	**0.6042**	0.4792
leukemia	0.9167	**1.0000**	0.9583	**1.0000**
madelon	0.5617	**0.5833**	**0.6083**	0.5833
ohsumed	0.9275	**0.9438**	0.9437	**0.9438**
apcj-etiology	0.9793	**0.9879**	0.9872	**0.9879**
dorothea	**0.9613**	0.9371	0.9343	**0.9371**
thrombin	**0.9374**	0.7300	**0.9613**	0.7300
news20	**0.7755**	0.7354	**0.8276**	0.7354
url1	0.9627	**0.9765**	0.9744	**0.9765**
kdd10	**0.878**	0.8179	**0.8723**	0.8179
webspam	0.9532	**0.9617**	0.9611	**0.9617**
Ave rank	1.3929	**1.6071**	1.3929	**1.6071**
w/t/l	-	5/4/5	-	5/6/3

Since GDM is implemented in C++, we developed a C++ version of SAOLA for the comparison with GDM, in addition to its Matlab version. In Figure 4, we only give the last four data sets with extremely high dimensionality in Table I, since on the the first ten data sets, the running time of both SAOLA and GDM is no more than ten seconds. We can see that both GDM and SAOLA are very efficient to handle extremely high-dimensional data sets. Except for the *news20* data set, SAOLA is a little faster than GDM. On the sparse data sets, SAOLA is faster than GDM, while on the dense data sets, such as the *news20* data set, GDM is faster than SAOLA. Finally, Figure 5 reports the number of selected features of SAOLA comparing to GDM. Except for the *breast-cancer* data set, SAOLA selects fewer features than GDM to achieve the very competitive prediction accuracy with GDM.

In summary, our SAOLA algorithm is a scalable and accurate online approach. This validates that without requiring a complete set of features on a training data set before feature

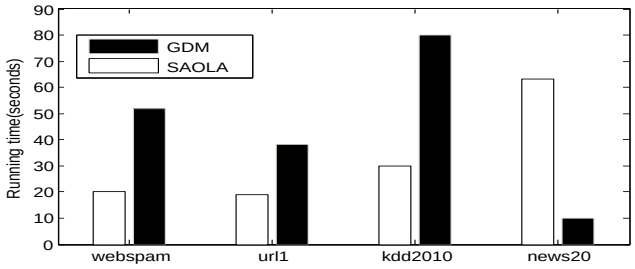

Fig. 4. Running time of SAOLA and GDM

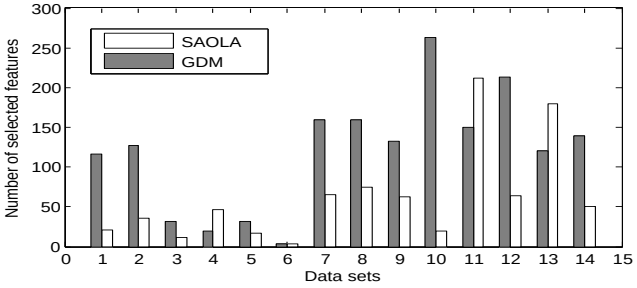

Fig. 5. Number of selected features (The labels of the x-axis from 1 to 10 denote the data sets: 1. dexter; 2. lung-cancer; 3. hiva; 4. breast-cancer; 5. leukemia; 6. madelon; 7. ohsumed; 8. apcj-etiology; 9. dorothea; 10. thrombin; 11. news20; 12. url1; 13. kdd10; 14. webspam)

selection starts, SAOLA is very competitive comparing to the well-established and state-of-the-art batch feature selection algorithms, FCBF, SPSF-LAR, and GDM.

D. Analysis of the Effect of Parameters

1) Analysis of Correlation Bounds: In Section III.B, we derived the correlation bound of $I(F_i;Y)$, that is, $\delta_2 = min(I(F_i;C), I(Y;C))$. To further validate the correlation bound δ_2 in Eq.(12) and Eq.(14), in this section, we conduct an empirical study by setting $\delta_2 = max(I(F_i;C), I(Y;C))$ in Algorithm 1, and derive a variant of the SAOLA algorithm, called the SAOLA-max algorithm. In the experiments, SAOLA-max uses the same parameters as SAOLA.

Table XII shows the prediction accuracies of SAOLA and SAOLA-max. With the summary of the win/tie/lose (w/t/l) counts in Table XII (paired t-tests at 95% significance level), we can see that SAOLA is very competitive with SAOLA-max in prediction accuracy. With the Friedman test at 95% significance level, under the null-hypothesis, which states that the performance of SAOLA and that of SAOLA-max have no difference, for the KNN classifier, the null-hypothesis cannot be rejected. The average ranks calculated from the Friedman test for SAOLA and SAOLA-max are 1.4643 and 1.5357, respectively. Meanwhile, with respect to J48, the average ranks for SAOLA and SAOLA-max are 1.4286 and 1.5714, respectively. The Friedman test testifies that SAOLA and SAOLA-max have no significant difference in prediction accuracy, although SAOLA-max gets the higher average ranks.

TABLE XII. PREDICTION ACCURACY

Dataset	KNN		J48	
	SAOLA	SAOLA-max	SAOLA	SAOLA-max
dexter	0.7600	0.8000	0.8133	**0.8300**
lung-cancer	**0.9833**	0.9500	**0.9500**	**0.9500**
hiva	**0.9635**	0.9505	**0.9661**	0.9557
breast-cancer	0.6771	**0.6875**	0.6042	**0.6458**
leukemia	0.9167	**1.0000**	**0.9583**	**0.9583**
madelon	**0.5617**	**0.5617**	**0.6083**	**0.6083**
ohsumed	**0.9275**	0.9256	**0.9437**	**0.9437**
apcj-etiology	0.9793	**0.9807**	**0.9872**	0.9870
dorothea	**0.9613**	0.9171	**0.9343**	0.9257
thrombin	0.9374	**0.9484**	**0.9613**	0.9503
news20	**0.7755**	0.7592	0.8276	**0.8295**
url1	0.9627	**0.9732**	0.9744	**0.9761**
kdd2010	**0.8780**	0.8766	0.8723	**0.8751**
webspam	0.9532	**0.9546**	0.9611	**0.9635**
Ave rank	1.4643	**1.5357**	1.4286	**1.5714**
w/t/l	-	4/5/5	-	2/10/2

However, on the running time, Table XIII shows that SAOLA is much more efficient than SAOLA-max on all data sets, especially on those of extremely high dimensionality. In Table XIII, we can also see that SAOLA selects fewer features than SAOLA-max. The explanation is that SAOLA-max uses a bigger relevance threshold ($\delta_2 = max(I(X;C), I(Y;C))$ for removing redundant features than SAOLA ($\delta_2 = min(I(X;C), I(Y;C))$. Clearly, the larger the relevance threshold δ_2, more features are added to the current feature set (see Steps 9 and 13 of Algorithm 1).

Compared to SAOLA-max, we can conclude that it is accurate and scalable to use the correlation bound, $\delta_2 = min(I(X;C), I(Y;C)$ in the SAOLA algorithm, for pairwise comparisons to filter out redundant features.

TABLE XIII. RUNNING TIME AND NUMBER OF SELECTED FEATURES

Dataset	Running time (seconds)		Number of selected features	
	SAOLA	SAOLA-max	SAOLA	SAOLA-max
dexter	**3**	3	**21**	39
lung-cancer	**6**	62	**35**	260
hiva	**1**	3	**12**	58
breast-cancer	**5**	40	**46**	93
leukemia	**2**	4	**17**	70
madelon	**0.1**	**0.1**	**3**	**3**
ohsumed	**6**	8	**65**	89
apcj-etiology	**22**	38	**75**	105
dorothea	**58**	327	**63**	516
thrombin	**63**	497	**20**	498
news20	**944**	2100	**212**	449
url1	**200**	526	**64**	346
kdd2010	**1056**	2651	**180**	193
webspam	**1456**	11606	**51**	165

2) The Effect of Input Order of Features: Since the dimensions are presented in a sequential scan, does the input order of the features have an impact on the quality of the selected feature set? To evaluate the effect on the SAOLA algorithm, we generate a number of trials in which each trial represents a random ordering of the features in the input feature set. We apply the SAOLA algorithm to each randomized trial and report the results in Figures 6 to 7, where the x-axis represents the randomized trials and the y-axis represents prediction accuracies of the corresponding trials. On the last eight very high-dimensional data sets, the results in Figures 6 to 7 confirm that varying the order of the incoming features does not affect much the final outcomes. Our explanation is that with various feature orders, Steps 9 and 13 of Algorithm 1 can select the feature with the highest correlation with the class attribute among a set of correlated features and remove the corresponding correlated features of this feature.

The only difference is that in some feature orders, the final feature subset may include some weakly relevant features. For example, assuming at time t, F_i arrives and has only one feature Y that satisfies Eq.(12) in the input features, and Y arrived before F_i and has stayed in the currently selected feature set $S^*_{t_{i-1}}$. Then F_i can be removed at time t given Y. But if F_i arrives before Y, and Y is removed before F_i's arrival, F_i cannot be removed later and may be kept in the final feature set. This also explains why there is a little fluctuation of prediction accuracy in each input order of features in Figures 6 and 7.

3) The Effect of Relevance Thresholds: The SAOLA algorithm has two versions: SAOLA with information gain for discrete data and SAOLA with the Fisher's Z-test for continuous data. For both versions, SAOLA needs to set a relevance threshold (δ_1 in Algorithm 1) in advance to determine whether two features are relevant. For discrete data, we set 11 different relevance thresholds for SAOLA on the *dorothea* and *thrombin* data sets. From Figure 8, we can see that in the term of prediction accuracy, the relevance thresholds do not have an significant impact on the SAOLA algorithm.

For the Fisher's Z-test, the relevance threshold is the significance level, α, and is always set to 0.01 or 0.05. Table XIV shows the results of SAOLA under the different significance

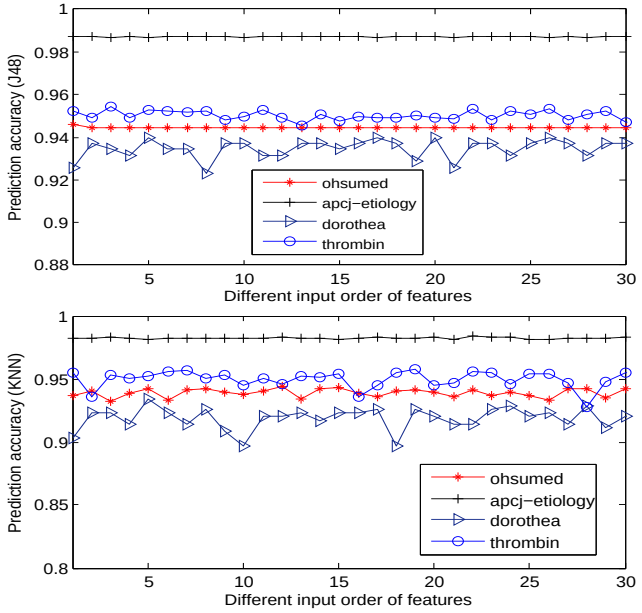

Fig. 6. Prediction accuracies on varied input orders of features (the top figure with J48 while the bottom figure with KNN)

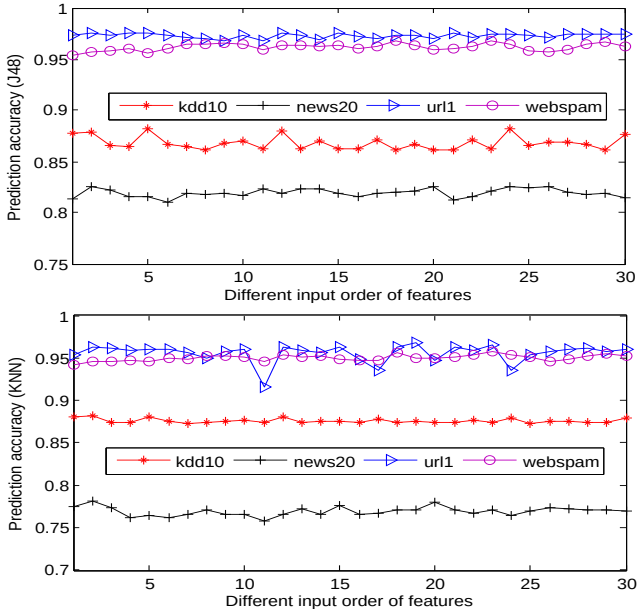

Fig. 7. Prediction accuracies on varied input orders of features (the top figure with J48 while the bottom figure with KNN)

TABLE XIV. PREDICTION ACCURACIES UNDER DIFFERENT SIGNIFICANCE LEVELS

Dataset	0.01/0.05(KNN)	0.01/0.05(J48)
ohsumed	0.9275/0.9394	0.9437/0.9437
apcj-etiology	0.9793/0.9838	0.9872/0.9873
news20	0.7755/0.7749	0.8276/0.8276
url1	0.9627/0.9642	0.9744/0.9744
kdd2010	0.8780/0.8678	0.8723/0.8723
webspam	0.9532/0.9493	0.9611/0.9611

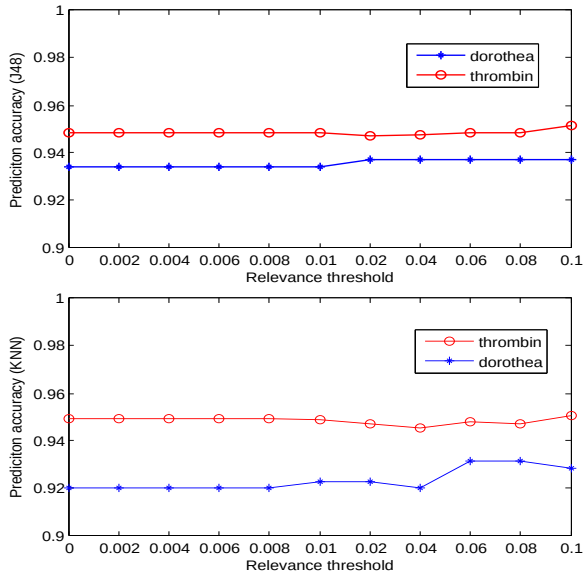

Fig. 8. Prediction accuracies on varied relevance thresholds (the top figure with J48 while the bottom figure with KNN)

levels. It is clear that a significant level does not impose a significant impact on the SAOLA algorithm either.

V. CONCLUSIONS

In this paper, we presented the SAOLA algorithm, a scalable and accurate online approach to tackle feature selection with extremely high dimensionality in a sequential scan. We conducted a theoretical analysis and derived a low bound of correlations between features for pairwise comparisons, and then proposed a set of online pairwise comparisons to maintain a parsimonious model over time in an online manner.

Using a series of benchmark real data sets, we compared the SAOLA algorithm with three state-of-the-art online feature selection methods and three batch feature selection algorithms. Our empirical study demonstrated that SAOLA is scalable on data sets of extremely high dimensionality, has superior performance over the three state-of-the-art online feature selection methods, and is very competitive with the three state-of-the-art batch feature selection methods in accuracy, while much faster in running time.

ACKNOWLEDGMENTS

This work is partly supported by a PIMS Post-Doctoral Fellowship Award of the Pacific Institute for the Mathematical Sciences, Canada, the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China (under grant IRT13059), the National 973 Program of China (under grant 2013CB329604), the National Natural Science Foundation of China (under grant 61229301), an NSERC Discovery grant and a BCIC NRAS Team Project. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11:171–234, 2010.

[2] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13:27–66, 2012.

[3] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[4] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[5] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[6] D. Koller and M. Sahami. Toward optimal feature selection. In *ICML-1995*, pages 284–292, 1995.

[7] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, 2005.

[8] J. M. Peña. Learning gaussian graphical models of gene networks with false discovery rate control. In *EvoBIO-2008*, pages 165–176. 2008.

[9] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[10] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13:1393–1434, 2012.

[11] M. Tan, L. Wang, and I. W. Tsang. Learning sparse svm for feature selection on very high dimensional datasets. In *ICML-2010*, pages 1047–1054, 2010.

[12] J. Tang and H. Liu. Unsupervised feature selection for linked social media data. In *ACM SIGKDD-2012*, pages 904–912. ACM, 2012.

[13] D. Wang, D. Irani, and C. Pu. Evolutionary study of web spam: Webb spam corpus 2011 versus webb spam corpus 2006. In *CollaborateCom-2012*, pages 40–49, 2012.

[14] J. Wang, P. Zhao, S. C. Hoi, and R. Jin. Online feature selection and its applications. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–14, 2013.

[15] A. Woznica, P. Nguyen, and A. Kalousis. Model mining for robust feature selection. In *ACM SIGKDD-2012*, pages 913–921. ACM, 2012.

[16] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu. Online feature selection with streaming features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1178–1192, 2013.

[17] L. Yu, C. Ding, and S. Loscalzo. Stable feature selection via dense feature groups. In *ACM SIGKDD-2008*, pages 803–811. ACM, 2008.

[18] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, 2004.

[19] Y. Zhai, M. Tan, I. Tsang, and Y. S. Ong. Discovering support and affiliated features from very high dimensions. In *ICML-2012*, pages 1455–1462, 2012.

[20] Z. Zhao, L. Wang, H. Liu, and J. Ye. On similarity preserving feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 25:619–632, 2013.

[21] J. Zhou, D. P. Foster, R. A. Stine, and L. H. Ungar. Streamwise feature selection. *Journal of Machine Learning Research*, 7:1861–1885, 2006.